WHAT IS CLAIMED IS:

1.   In a computing environment, a method comprising:

receiving an original model comprising a plurality of

elements representing operations to test;

5   producing a plurality of subsequences from the elements,

each subsequence corresponding to at least two elements; and

generating a suite of test cases from the subsequences,

such that any valid subsequence appears at least once among

the test cases in the suite.

10

2.   The method of claim 1 further comprising, evaluating

a constraint against the test cases to determine for each test

case whether that test case matches the constraint.

15   3.   The method of claim 2 wherein the constraint matches

a selected test case, and further comprising, splitting the

test case into at least two test cases.

4.   The method of claim 1 further comprising, evaluating

20   a precondition against the test cases to determine for each

test case whether that test case matches the precondition.

5.   The method of claim 2 wherein the precondition matches a selected test case, and further comprising, splitting the test case into at least two test cases.

5   6.   The method of claim 1 wherein a plurality of preconditions are known, and further comprising, sorting the preconditions into an order, and evaluating each precondition against the test cases based on the order to determine for each test case whether that test case matches the

10   precondition.

7.   The method of claim 1 wherein generating a suite of test cases from the subsequences comprises selecting a subsequence based on a selection algorithm, and adding the

15   selected subsequence to a test case.

8.   The method of claim 7 further comprising, marking the selected subsequence as covered.

20   9.   The method of claim 8 further comprising, selecting another subsequence from a set of uncovered subsequences, and adding the other subsequence to the test case.

10. The method of claim 8 wherein selecting a
subsequence based on a selection algorithm comprises,
determining which element starts a largest number of still
uncovered subsequences, and selecting a subsequence starting
5   with that element.

11. The method of claim 8 wherein selecting a
subsequence based on a selection algorithm comprises,
determining which element starts a largest number of still
10 uncovered subsequences, and if there is only one such an
element, selecting a subsequence starting with that element,
and if there is a tie, employing a tiebreaker.

12. The method of claim 1 wherein generating a suite of
15 test cases from the subsequences comprises selecting a
subsequence, adding the selected subsequence to a test case,
marking the selected subsequence as covered, and repeating
until no subsequence remains uncovered.

20   13. A computer-readable medium having computer-
executable instructions for performing the method of claim 1.

- 42 -

14. In a computing environment, a system comprising:

means for receiving a model file comprising a plurality of elements representing operations to test;

means for producing a plurality of subsequences from the elements, each subsequence corresponding to at least two elements; and

means for generating a suite of test cases from the subsequences, including means for selecting a subsequence, means for adding the selected subsequence to a test case, means for marking the selected subsequence as covered, such that any valid subsequence is covered at least once among the test cases in the suite.

15. The system of claim 14 further comprising, evaluation means for evaluating a constraint against the test cases to determine for each test case whether that test case matches the constraint, and if so, the evaluation means employing means for splitting that test case into at least two test cases.

16. The system of claim 14 further comprising, evaluation means for evaluating a precondition against the test cases to determine for each test case whether that test case matches the constraint, and if so, the evaluation means

employing means for splitting that test case into at least two test cases.

17.   The system of claim 14 wherein the means for selecting a subsequence comprises means for counting which element starts a still most uncovered number of subsequences.

18.   In a computing environment, a system for constructing test cases, comprising:

a model file reader that receives a model file containing elements representing operations to test in various sequences;

a pre-processing mechanism that arranges the elements into a set of subsequences, each subsequence in the set of the same length and comprising at least two elements;

a selection mechanism that selects a subsequence from the set of subsequences, adds the subsequence to a test case, and marks the subsequence as covered; and

the selection mechanism controlled to add subsequences to test cases until no subsequence remains uncovered.

19.   The system of claim 18 further comprising a post-processing mechanism that is configured to split a single test case into at least two test cases based on any preconditions or constraints associated with the model file.

20.   The system of claim 18 wherein the selection

mechanism employs a selection algorithm to select a

subsequence, by determining which element starts a largest

5   number of still uncovered subsequences, and if there is only

one such an element, selecting a subsequence starting with

that element, and if there is a tie, employing a tiebreaker.

21.   The system of claim 18 wherein the tiebreaker

10   comprises selecting a first available subsequence.

22.   The method of claim 20 wherein the selection

mechanism adds another subsequence to the test set based on a

subsequence already in the test set.

15

23.   The method of claim 22 wherein the selection

mechanism first looks for the other subsequence among

uncovered subsequences in the set, and if none can be used,

secondarily looks for a covered subsequence in the set.

20